

SYSTEM AND METHOD FOR DYNAMIC SECURITY PROVISIONING OF COMPUTING RESOURCES

DESCRIPTION

CROSS-REFERENCE TO RELATED APPLICATION

[Para 1] This application claims priority to, and the benefit of, U.S. Provisional Application Serial No. 60/521,498 filed May 6, 2004 and entitled "System and Method for Dynamic Security Provisioning in a Commodity Computing Environment", which is hereby incorporated by reference.

FIELD OF INVENTION

[Para 2] The present invention generally relates to provisioning and de-provisioning of computing resources for an asset within a commodity computing environment, and more particularly, to a system and method for dynamic provisioning of computing resources for assets through a multi-dimensional decisioning process including asset classification, asset business value and requesting entity.

BACKGROUND OF INVENTION

[Para 3] In enterprise computing environments where computing resources may be shared between any number of applications, processes, data transactions, etc., the classic problem that frequently arises is how to allocate

limited computing resources to various requesting entities which may include, for example, a person, an application, a process, a server and the like. In recent years, this problem has become more manageable with the development of higher power processors and multi-processor computing environments with load-balancing capabilities. However, in high traffic computing environments, maintaining an adequate level of computing resources to adequately manage periods of high volume demand can often be costly. Further, adding additional computing resources is often not cost effective as it may result in extended periods where resources are not sufficiently employed. For example, while all computing resources may be employed during periods of high-traffic which represents 23% of total computing time, many of those assets may be unused for the remaining 77% of total computing time. Efforts, such as load balancing, has resulted in some alleviation of the problems associated with limited resources during high demand, however other problems still exist.

[Para 4] Load balancing is a process by which a load (number of requests, number of users, etc.) is spread throughout a network of servers and other devices in order to prevent individual devices from becoming overwhelmed by traffic, which may cause it to fail or to process the traffic more slowly. Load balancing may also be employed to redirect traffic in the event of a server or device failure thereby providing a level of fault tolerance. While load balancing is effective in managing computing resources in accordance with varying traffic levels, the technology is generally unconcerned with the nature of the traffic itself. In other words, load balancing utilities, as employed within most enterprises, do not apply security controls to computing assets beyond a one-dimensional decision point which is typically data classification.

[Para 5] Therefore, a need exists for a system and method for application of security controls and dynamic provisioning of computing resources that takes into account multi-dimensional decision criteria based on asset classification, business value and resource classification for a requesting entity.

SUMMARY OF INVENTION

[Para 6] The invention provides for dynamic security provisioning within a commodity computing environment. The invention implements multi-dimensional decision criteria to assign an asset to an appropriate security domain. In general, the invention determines a classification for an asset, determines a business value for the asset and/or determines a resource classification for the requesting entity. A first dimension may describe a classification for data, system, application, process and hardware assets. For example, a classification may be labeled public, business confidential, private, or secret. A second dimension may provide a business' value assessment of an asset such as, for example, low, medium or high. A third dimension may provide a resource classification of the entity attempting to access the asset. Such asset classifications may include, for example, a label of "trusted" for internal entities or "non-trusted" for external entities. The multi-dimensional decisioning factor may create a set of security domains where assets may be placed; each comprising varying degrees of security controls.

[Para 7] A security domain effects how computing resources will be provisioned or de-provisioned in response to a security domain's asset. In a commodity computing environment, a security domain indicative of classification and value indices of an asset enables logic to be applied dictating what types of resources will be allocated to the asset and at what level. Assets residing in a security domain that is considered to be critical, will be afforded prioritized provisioning over assets residing in a non-critical security domain. Further, the security domain may define how an asset is to be handled. For example, if the classification of a requesting entity is "un-trusted", then an asset may be placed in a security domain requiring additional authentication of the requesting entity.

[Para 8] A provisioning engine may define a provisioning configuration for an asset through inputs from a variety of manager components. The

manager components may have very specialized functionality and may include, for example, a connection manager, policy manager, configuration manager, risk manager, recovery manager, etc. Policies regarding a security domain may be processed by the provisioning engine which then transmits instructions to a provisioning tool which configures system resources for the asset. Further, to ensure that system resources are not unnecessarily reserved, the provisioning engine may send instructions to a provisioning tool to de-provision system resources when they are no longer needed by the asset or following a defined period of time.

[Para 9] More particularly, the invention includes a computer implemented method for dynamically provisioning computing resources. The method may include, for example, determining an asset classification, receiving a request for computing resources, wherein the request is associated with an asset, business value and/or resource classification related to the asset; and, provisioning the computing resources based on the determining step. Based on multi-dimensional decision criteria, the invention may also place assets in at least one of a plurality of security domains based on the determining step, wherein each security domain includes a different degree of security controls. The invention may then further provision computing resources based upon the security domain placement.

BRIEF DESCRIPTION OF THE DRAWINGS

[Para 10] A more complete understanding of the present invention may be derived by referring to the detailed description and claims when considered in connection with the Figures, wherein like reference numbers refer to similar elements throughout the Figures, and:

[Para 11] Figure 1 is a block diagram illustrating exemplary major system components for asset based provisioning in a commodity computing environment;

[Para 12] Figure 2 is a block diagram illustrating an exemplary relationship between a provisioning engine and various resource management components;

[Para 13] Figure 3 is a flow chart illustrating an exemplary method for classifying an asset into an appropriate security domain; and,

[Para 14] Figure 4 is a flow chart illustrating an exemplary method for providing access to an asset and provisioning and/or de-provisioning computing and data resources.

DETAILED DESCRIPTION

[Para 15] The detailed description of exemplary embodiments herein makes reference to the accompanying drawings, which show the exemplary embodiment by way of illustration and its best mode. While these exemplary embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, it should be understood that other embodiments may be realized and that logical and mechanical changes may be made without departing from the spirit and scope of the invention. Thus, the detailed description herein is presented for purposes of illustration only and not of limitation.

[Para 16] In general, the invention includes a system and method for dynamic provisioning of computing resources. The invention may be applicable in a commodity computing environment. Through connections with various management utilities 135, the provisioning engine 130 may consider multiple levels of indices relating to an asset to ensure proper and efficient allocation of computing resources. Based on an asset's security domain, a provisioning engine 130 may define baseline security requirements, dictate what level of resources the asset will be afforded and interface with a system management tool 145 to provision and de-provision system resources.

[Para 17] The security provisioning system 100 contains various hardware and/or software components suitably configured to perform analysis of incoming asset requests and asset attributes. Security provisioning system 100 may apply appropriate security controls and provision system resources accordingly. In one embodiment, with reference to Figure 1, security provisioning system 100 includes a firewall 115, an Internet server 120, an applications server 125, provisioning engine 130, management utilities 135, domain database 140 and management tools 145. External entities may include an external client 105 (e.g., Internet client) and an internal client 110 (e.g., a Local Area Network (LAN) or Wide Area Network (WAN) client). The security provisioning system 100 may interface with any number of computing devices and data in one or more enterprise 150. An enterprise 150 may comprise any number of mainframes, servers, applications and databases. For simplicity, the enterprise 150 is represented by a mainframe computer 155, accounts payable server 160 and a database 165.

[Para 18] As will be appreciated by one of ordinary skill in the art, the invention may be embodied as a customization of an existing system, an add-on product, upgraded software, a distributed system, a method, a data processing system, a device for data processing, and/or a computer program product. Accordingly, the present invention may take the form of an entirely software embodiment, an entirely hardware embodiment, or an embodiment combining aspects of both software and hardware. Furthermore, the present invention may take the form of a computer program product on a computer-readable storage medium having computer-readable program code means embodied in the storage medium. Any suitable computer-readable storage medium may be utilized, including hard disks, CD-ROM, optical storage devices, magnetic storage devices, and/or the like.

[Para 19] Referring again to Figure 1, an external client 105 may include any individual, business, entity, software and/or hardware that utilizes applications and or data within an enterprise 150. For example, an external client 105 may take the form of a bank customer who connects to a bank's website in order to conduct bill payment and transfer transactions. Such

transactions may comprise requests for various enterprise system 150 computing assets such as, for example, applications, processes and data. The manner in which requested assets will be provisioned may have been determined by the classifications of the assets according to multi-dimensional decision criteria. An external client 105 may be equipped with an Internet browser connected to the Internet using standard dial-up, cable, DSL or any other Internet protocol known in the art. Transactions originating at an external client 105 may need to pass through a firewall 115 in order to prevent unauthorized access from other networks. Further, the provisioning engine may manage external client 105 connections by invoking a firewall 115 or allowing direct access to an Internet server 120.

[Para 20] A firewall 115 may include any hardware and/or software suitably configured to protect security provisioning system 100 components and/or enterprise computing resources from users of other networks. Further, a firewall may be configured to limit or restrict access to various systems and components behind the firewall 115 for external clients 105 connecting through an Internet server 120. Firewall 115 may reside in varying configurations including Stateful Inspection, Proxy based and Packet Filtering among others. Firewall 115 may be integrated within an Internet server 120 or any other security provisioning system 100 components or may further reside as a separate entity.

[Para 21] An external client 105 and an internal client 110 may be equipped with computing units or systems which may take the form of a computer or set of computers, although other types of computing units or systems may be used, including laptops, notebooks, hand held computers, set-top boxes, workstations, computer-servers, main frame computers, mini-computers, PC servers, network sets of computers, and/or the like. Further, any computing device discussed herein may also be a “pervasive computing device,” which may include a traditionally non-computerized device that is embedded with a computing unit. Examples can include, for example, watches, Internet enabled kitchen appliances, restaurant tables embedded with RF readers, wallets or purses with imbedded transponders, etc.

[Para 22] In one embodiment, an internal client 110 may be an administrator or manager of the provisioning system 100 and/or enterprise 150. The internal client 110 may interface with the provisioning system 100 to configure management utilities 135, define classification rules, override provisioning engine 130, monitor processes and the like.

[Para 23] An Internet server 120 may include any hardware and/or software suitably configured to serve web pages and data to a connected external client 105. The Internet server 120 may communicate with any number of other servers, databases or components through any means known in the art. The Internet server 120 may serve as a conduit between an external client 105 and the systems and components of the security provisioning system 100 and the enterprise 150. Internet server 120 may interface with the provisioning engine 130 where requests for assets may invoke a provisioning process.

[Para 24] An asset, as used herein, may include any hardware, software and/or data components which perform general and/or specific functions within an enterprise computing environment. An asset may be a component that is shared between any number of users and/or other computing assets. An asset may comprise hardware components (e.g., computer memory), software and/or data components (e.g., computer programs, code components, processes and data). Like tangible assets such as, for example, office furniture and supplies, computing assets may be assigned values according to the importance of their utility. Further, the usability of an asset may be directly related to the availability of computing resources to process the asset.

[Para 25] An applications server 125 may include any hardware and/or software suitably configured to serve applications and data to a connected internal client 110. Like the Internet server 120, the applications server 125 may communicate with any number of other servers, databases and/or components through any means known in the art. Further, the applications server 125 may serve as a conduit between an internal client 110 and the systems and components of the security provisioning system 100 and the

enterprise 150. Internal clients 110 may interface with an applications server 125 through any means known in the art including a LAN/WAN, for example.

[Para 26] As used herein, Internet Server 120 and applications server 125 may be used interchangeably to illustrate a connection point for both internal clients 110 and external clients 105. While an Internet server 120 may not be necessary for internal clients 110 to interface with the various systems and components of the security provisioning system 100 and the enterprise 150, it is shown in Figure 1 to demonstrate an alternative system and method for accessing the utility of the invention. Practitioners will appreciate that there may be any number of architectures and configurations, including hardware, software and networking protocols, for interfacing with enterprise 150 resources.

[Para 27] The provisioning engine 130 may include any hardware and/or software suitably configured to receive inputs from one or more manager components and transmit instructions to management tools 145 regarding provisioning and de-provisioning of enterprise 150 computing resources. The provisioning engine 130 may reside as a software entity within an Internet server 120 or applications server 125. The provisioning engine 130 may also reside as a stand-alone system within the security provisioning system 100. Provisioning engine 130 may receive requests for assets from an Internet server 120 or applications server 125. Based on such requests, the provisioning engine 130 may solicit inputs from a number of management utilities 135 regarding provisioning rules based on the asset's security. In one embodiment, a provisioning engine 130 may request data from a domain database 140 which may maintain security domain definitions and policies.

[Para 28] Domain database 140 may include any hardware and/or software suitably configured to facilitate storing security domain information relating to various security tiers and used to maintain security domain definitions and policies relating to the domains. One skilled in the art will appreciate that the security provisioning system 100 may employ any number of databases in any number of configurations. Further, any databases discussed herein may be any type of database, such as relational, hierarchical,

graphical, object-oriented, and/or other database configurations. Common database products that may be used to implement the databases include DB2 by IBM (White Plains, NY), various database products available from Oracle Corporation (Redwood Shores, CA), Microsoft Access or Microsoft SQL Server by Microsoft Corporation (Redmond, Washington), or any other suitable database product. Moreover, the databases may be organized in any suitable manner, for example, as data tables or lookup tables. Each record may be a single file, a series of files, a linked series of data fields or any other data structure. Association of certain data may be accomplished through any desired data association technique such as those known or practiced in the art. For example, the association may be accomplished either manually or automatically. Automatic association techniques may include, for example, a database search, a database merge, GREP, AGREP, SQL, and/or the like. The association step may be accomplished by a database merge function, for example, using a “key field” in pre-selected databases or data sectors.

[Para 29] More particularly, a “key field” partitions the database according to the high-level class of objects defined by the key field. For example, certain types of data may be designated as a key field in a plurality of related data tables and the data tables may then be linked on the basis of the type of data in the key field. In this regard, the data corresponding to the key field in each of the linked data tables is preferably the same or of the same type. However, data tables having similar, though not identical, data in the key fields may also be linked by using AGREP, for example. In accordance with one aspect of the present invention, any suitable data storage technique may be utilized to store data without a standard format. Data sets may be stored using any suitable technique, including, for example, storing individual files using an ISO/IEC 7816-4 file structure; implementing a domain whereby a dedicated file is selected that exposes one or more elementary files containing one or more data sets; using data sets stored in individual files using a hierarchical filing system; data sets stored as records in a single file (including compression, SQL accessible, hashed via one or more keys, numeric, alphabetical by first tuple, etc.); binary large object (BLOB); stored as ungrouped data elements encoded using ISO/IEC 7816-6 data elements;

stored as ungrouped data elements encoded using ISO/IEC Abstract Syntax Notation (ASN.1) as in ISO/IEC 8824 and 8825; and/or other proprietary techniques that may include fractal compression methods, image compression methods, etc.

[Para 30] In one exemplary embodiment, the ability to store a wide variety of information in different formats is facilitated by storing the information as a binary large object (BLOB). Thus, any binary information can be stored in a storage space associated with a data set. As discussed above, the binary information may be stored on the financial transaction instrument or external to but affiliated with the financial transaction instrument. The BLOB method may store data sets as ungrouped data elements formatted as a block of binary via a fixed memory offset using either fixed storage allocation, circular queue techniques, or best practices with respect to memory management (e.g., paged memory, least recently used, etc.). By using BLOB methods, the ability to store various data sets that have different formats facilitates the storage of data associated with the financial transaction instrument by multiple and unrelated owners of the data sets. For example, a first data set which may be stored may be provided by a first issuer, a second data set which may be stored may be provided by an unrelated second issuer, and yet a third data set which may be stored, may be provided by an third issuer unrelated to the first and second issuer. Each of these three exemplary data sets may contain different information that is stored using different data storage formats and/or techniques. Further, each data set may contain subsets of data which also may be distinct from other subsets.

[Para 31] As stated above, in various embodiments of the present invention, the data can be stored without regard to a common format. However, in one exemplary embodiment of the present invention, the data set (e.g., BLOB) may be annotated in a standard manner when provided for manipulating the data onto the financial transaction instrument. The annotation may comprise a short header, trailer, or other appropriate indicator related to each data set that is configured to convey information useful in managing the various data sets. For example, the annotation may be called a

“condition header”, “header”, “trailer”, or “status”, herein, and may comprise an indication of the status of the data set or may include an identifier correlated to a specific issuer or owner of the data. In one example, the first three bytes of each data set BLOB may be configured or configurable to indicate the status of that particular data set; e.g., LOADED, INITIALIZED, READY, BLOCKED, REMOVABLE, or DELETED. Subsequent bytes of data may be used to indicate for example, the identity of the issuer, user, transaction/membership account identifier or the like. Each of these condition annotations are further discussed herein.

[Para 32] The data set annotation may also be used for other types of status information as well as various other purposes. For example, the data set annotation may include security information establishing access levels. The access levels may, for example, be configured to permit only certain individuals, levels of employees, companies, or other entities to access data sets, or to permit access to specific data sets based on the transaction, provider, issuer, user or the like. Furthermore, the security information may restrict/permit only certain actions such as accessing, modifying, and/or deleting data sets. In one example, the data set annotation indicates that only the data set owner or the user are permitted to delete a data set, various identified providers are permitted to access the data set for reading, and others are altogether excluded from accessing the data set. However, other access restriction parameters may also be used allowing various entities to access a data set with various permission levels as appropriate.

[Para 33] The data, including the header or trailer may be received by a stand alone interaction device configured to add, delete, modify, or augment the data in accordance with the header or trailer. As such, in one embodiment, the header or trailer is not stored on the transaction device along with the associated issuer-owned data but instead the appropriate action may be taken by providing to the transaction instrument user at the stand alone device, the appropriate option for the action to be taken. The present invention may contemplate a data storage arrangement wherein the header or trailer, or

header or trailer history, of the data is stored on the transaction instrument in relation to the appropriate data.

[Para 34] Management utilities 135 may include any hardware and/or software suitably configured to manage and enforce business rules and policies relating to access to assets and provisioning of computing resources. Management utilities 135 may comprise various components with specific functionality such as, for example, a policy manager, connection enforcement manager, risk manager, configuration manager, recovery manager, etc. While illustrated as a single component in Figure 1, management utilities 135 may be deployed as individual components residing on a single server or computing unit, or may reside within any number of computing components and/or devices. When the provisioning engine 130 receives an asset request from an Internet server 120 or server 125, it may invoke one or more management utilities 135 in order to retrieve policy information which may be used to define a provisioning configuration. Based on input from the one or more management utilities 135, the provisioning engine may create a security domain for the asset in order to safely and efficiently provision computing resources and grant access to the requested asset. A more detailed description of the exemplary management utilities 135 will be provided herein.

[Para 35] Management tools 145 may include any hardware and/or software suitably configured to manage memory and processor allocation in response to instructions provided by the provisioning engine 130. Those skilled in the art will appreciate that there are a number of methods known in the art for dynamically managing computing resources including solutions which are commercially available such as Tivoli™ by IBM®. Such management tools may vary in sophistication and be configured for any number of computing architectures and operating systems. The provisioning engine 130 may interface with management tools 145 in order to provide instructions regarding resource provisioning and de-provisioning for computing systems residing in the enterprise 150.

[Para 36] An enterprise 150 may include any hardware and/or software suitably configured to carry out computing operations for a business,

government entity or organization. As used herein, the enterprise 150 is used to describe one or more computing systems, which may include servers, databases, mainframes, printers and the like. An enterprise 150 may provide access to varying levels applications and data to internal clients 110, external clients 105 as well as to any other computing device and/or component. Various computing systems within the enterprise 150 may be invoked in response to a request for application and data processing. This request may come from another enterprise 150 component, a security provisioning system 100 component or a combination thereof. However, in most cases, the request will have originated from a user, such as the external client 105 or internal client 110.

[Para 37] Figure 2 is a block diagram illustrating the relationship between an exemplary provisioning engine of the invention and various resource manager components. Block 200 represents a request for an asset which may originate from an external client 100 or internal client 110. When the provisioning engine 205 receives a request related to an asset 200, it may call on one or more manager components to assess the asset and the requesting entity to provide the provisioning engine 205 with the information required to apply appropriate security controls and provisioning based on an asset's security domain. In one embodiment, decisions regarding the security domain and provisioning thereof may generally be applied according to pre-defined business rules or policies regarding the asset and requesting entity. Such policies may be stored within a database or file.

[Para 38] On receiving a request 200, the provisioning engine 205 may invoke a risk manager 210 in order to ensure that the connection from which the request 200 has originated is not a risk to the enterprise 150. For example, a risk manager 210 may examine the operating system of the requesting client to ensure that the latest software patch has been applied. Patches are code components that are most often released by the manufacturer of hardware and/or software products to correct problems that were not recognized prior to release of the products. Further, the risk manager 210 may verify that the requesting client has anti-virus software and

that the most recent patch level has been installed. In today's computing environment where security is critical, patches are often released to reduce a computer system's vulnerability to malicious code. Computing systems that do not have the appropriate protection can inadvertently and unknowingly download such malicious code, otherwise known as worms, viruses and attacks. Subsequent connections with other computing systems may result in "infection" of those systems by the affected computer.

[Para 39] The risk manager 210 may report information regarding the "health" of the requesting client to a connection manager 225 which may contain connection policies defining which connections to allow, limit, or decline. If the information from the risk manager 210 indicates that the appropriate patches and/or virus protection are absent from the connecting client, the connection manager 225 may provide the requesting client a link to a website, for example, where they may download the required patches and/or virus protection software. A requesting client not conforming to the security policies of the connection manager 225, may be restricted from accessing an asset until the appropriate patch and or software has been installed on the client's computer.

[Para 40] The provisioning engine 205 may invoke a policy manager 215 in order to evaluate the nature of the requested asset. The policy manager 215 may evaluate information contained in an asset request in order to determine workload allocation decisioning. Based on this determination, policy information may be provided to determine the appropriate controls to apply to an asset. Such controls may comprise requirement definitions relating to authentication, authorization, confidentiality, integrity, availability and the like. For example, a policy manager may apply controls to determine whether an application requires firewall services or intrusion detection services. In one embodiment, such controls may be encoded in XML and acted upon by a policy enforcement aspect of the policy manager 215.

[Para 41] A configuration manager 220 may be invoked by the provisioning engine 205 to determine what a server might restrict with regard to which services are available. For example, in a more secure environment,

the configuration manager 220 may instruct the provisioning engine 205 to disable all services except for those which are absolutely necessary. A provisioning engine 205 may provision services to an asset based on input from a configuration manager 220 in light of an asset's security domain.

[Para 42] A recovery manager 230 may store rules and procedures regarding how requests are to be processed during resource downtime do to system failures or recovery efforts. For example, most enterprise computing environments are designed to include failsafe measures wherein if one system fails, there are one or more other systems capable of taking over the processing from the failed system. However, in the event of a catastrophic failure or a server failure during high traffic, there may not be sufficient backup processing capabilities to maintain business as usual. Therefore, decisions must be made as to what processes are critical and which are not. Non-critical processes may be terminated, while others may be directed to another processor. The recovery manager 230 may assess requests during such a failure in order to provide such input to the provisioning engine 205. The provisioning engine 205 may determine whether or not to allocate the requested resources for the asset based on its assigned security domain.

[Para 43] Several managers have been presented in Figure 2, however practitioners will appreciate that there are any number of manager components that may assist the provisioning engine 205 in classifying an asset into a security domain. For example, the invention may further include a regulatory or compliance manager which may evaluate a request in order to determine if the request is related to FCC requirements. A request for an asset based on regulatory rules or laws may be given special priority where limited computing resources exist.

[Para 44] Based on inputs from the manager components according to an asset's security domain, the provisioning engine 205 may assign varying levels of security controls and provisioning rules that apply to the asset. For example, if the asset represents sensitive information, such as unreleased earnings analysis, a provisioning engine 205 may require higher levels of authentication for the requesting entity. This could include adding an

additional layer to the standard user ID and password authentication, such as requiring a smartcard or biometric data. According to various policies relating to the security domain, the provisioning engine 205 may transmit instructions to a systems provisioning tool 235. The systems provisioning tool 235 may control computing systems at the processor level allowing it to provision and de-provision system resources 240 in accordance with provisioning engine 205 instructions.

[Para 45] The various system components discussed herein may include one or more of the following: a server or other computing systems including a processor for processing digital data; a memory coupled to said processor for storing digital data; an input digitizer coupled to the processor for inputting digital data; an application program stored in said memory and accessible by said processor for directing processing of digital data by said processor; a display device coupled to the processor and memory for displaying information derived from digital data processed by said processor; and a plurality of databases. Various databases used herein may include: user data, debt data, income data, provider data; financial institution data; and/or like data useful in the operation of the present invention. As those skilled in the art will appreciate, user computer may include an operating system (e.g., Windows NT, 95/98/2000, OS2, UNIX, Linux, Solaris, MacOS, etc.) as well as various conventional support software and drivers typically associated with computers. user computer can be in a home or business environment with access to a network. In an exemplary embodiment, access is through a network or the Internet through a commercially-available web-browser software package.

[Para 46] As used herein, the term "network" shall include any electronic communications means which incorporates both hardware and software components of such. Communication among the parties in accordance with the present invention may be accomplished through any suitable communication channels, such as, for example, a telephone network, an extranet, an intranet, Internet, point of interaction device (point of sale device, personal digital assistant, cellular phone, kiosk, etc.), online communications,

off-line communications, wireless communications, transponder communications, local area network (LAN), wide area network (WAN), networked or linked devices and/or the like. Moreover, although the invention is frequently described herein as being implemented with TCP/IP communications protocols, the invention may also be implemented using IPX, Appletalk, IP-6, NetBIOS, OSI or any number of existing or future protocols. If the network is in the nature of a public network, such as the Internet, it may be advantageous to presume the network to be insecure and open to eavesdroppers. Specific information related to the protocols, standards, and application software utilized in connection with the Internet is generally known to those skilled in the art and, as such, need not be detailed herein. See, for example, DILIP NAIK, INTERNET STANDARDS AND PROTOCOLS (1998); JAVA 2 COMPLETE, various authors, (Sybex 1999); DEBORAH RAY AND ERIC RAY, MASTERING HTML 4.0 (1997); and LOSHIN, TCP/IP CLEARLY EXPLAINED (1997) and DAVID GOURLEY AND BRIAN TOTTY, HTTP, THE DEFINITIVE GUIDE (2002), the contents of which are hereby incorporated by reference.

[Para 47] The various system components may be independently, separately or collectively suitably coupled to the network via data links which includes, for example, a connection to an Internet Provider (ISP) over the local loop as is typically used in connection with standard modem communication, cable modem, Dish networks, ISDN, Digital Subscriber Line (DSL), or various wireless communication methods. See, e.g., GILBERT HELD, UNDERSTANDING DATA COMMUNICATIONS (1996), hereby incorporated by reference. It is noted that the network may be implemented as other types of networks, such as an interactive television (ITV) network. Moreover, the system contemplates the use, sale or distribution of any goods, services or information over any network having similar functionality described herein.

[Para 48] The computers discussed herein may provide a suitable website or other Internet-based graphical user interface which is accessible by users, hosts or operators of the system. In one embodiment, the Microsoft Internet Information Server (IIS), Microsoft Transaction Server (MTS), and Microsoft SQL Server, are used in conjunction with the Microsoft operating

system, Microsoft NT web server software, a Microsoft SQL Server database system, and a Microsoft Commerce Server. Additionally, components such as Access or Microsoft SQL Server, Oracle, Sybase, Informix MySQL, InterBase, etc., may be used to provide an Active Data Object (ADO) compliant database management system.

[Para 49] Any of the communications, inputs, storage, databases or displays discussed herein may be facilitated through a website having web pages. The term “web page” as it is used herein is not meant to limit the type of documents and applications that might be used to interact with the user. For example, a typical website might include, in addition to standard HTML documents, various forms, Java applets, JavaScript, active server pages (ASP), common gateway interface scripts (CGI), extensible markup language (XML), dynamic HTML, cascading style sheets (CSS), helper applications, plug-ins, and the like. A server may include a web service which receives a request from a web server, the request including a URL (<http://yahoo.com/stockquotes/ge>) and an IP address (123.56.789). The web server retrieves the appropriate web pages and sends the data or applications for the web pages to the IP address. Web services are applications which are capable of interacting with other applications over a communications means, such as the Internet. Web services are typically based on standards or protocols such as XML, SOAP, WSDL and UDDI. Web services methods are well known in the art, and are covered in many standard texts. See, e.g., ALEX NGHIEM, IT WEB SERVICES: A ROADMAP FOR THE ENTERPRISE (2003), hereby incorporated herein by reference.

[Para 50] The present invention may be described herein in terms of functional block components, screen shots, optional selections and various processing steps. It should be appreciated that such functional blocks may be realized by any number of hardware and/or software components configured to perform the specified functions. For example, the present invention may employ various integrated circuit components, e.g., memory elements, processing elements, logic elements, look-up tables, and the like, which may carry out a variety of functions under the control of one or more microprocessors or other control devices. Similarly, the software elements of

the present invention may be implemented with any programming or scripting language such as C, C++, Java, COBOL, assembler, PERL, Visual Basic, SQL Stored Procedures, extensible markup language (XML), with the various algorithms being implemented with any combination of data structures, objects, processes, routines or other programming elements. Further, it should be noted that the present invention may employ any number of conventional techniques for data transmission, signaling, data processing, network control, and the like. Still further, the invention could be used to detect or prevent security issues with a client-side scripting language, such as JavaScript, VBScript or the like. For a basic introduction of cryptography and network security, the following may be helpful references: (1) "Applied Cryptography: Protocols, Algorithms, And Source Code In C," by Bruce Schneier, published by John Wiley & Sons (second edition, 1996); (2) "Java Cryptography" by Jonathan Knudson, published by O'Reilly & Associates (1998); (3) "Cryptography & Network Security: Principles & Practice" by William Stallings, published by Prentice Hall; all of which are hereby incorporated by reference.

[Para 51] Each participant in the present invention may be equipped with a computing device in order to interact with the system and various assets. As previously described, the external client 105 and internal client 110 may each have a computing unit in the form of a personal computer, although other types of computing units may be used including laptops, notebooks, hand held computers, set-top boxes, cellular telephones, touch-tone telephones and the like. In one embodiment, the internal client may have a computing unit implemented in the form of a computer-server, although other implementations are contemplated by the invention. The enterprise 150 computing center as shown may comprise one or more mainframe computers. However, the enterprise 150 computing center may be implemented in other forms, such as a mini-computer, a PC server, a network of computers located in the same of different geographic locations, or the like. Moreover, the system contemplates the use, sale or distribution of any services or information over any network having similar functionality described herein.

[Para 52] These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer-implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[Para 53] Accordingly, functional blocks of the block diagrams and flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions, and program instruction means for performing the specified functions. It will also be understood that each functional block of the block diagrams and flowchart illustrations, and combinations of functional blocks in the block diagrams and flowchart illustrations, can be implemented by either special purpose hardware-based computer systems which perform the specified functions or steps, or suitable combinations of special purpose hardware and computer instructions.

[Para 54] Referring now to Figures 3–4, the process flows depicted are merely exemplary embodiments of the invention and are not intended to limit the scope of the invention as described herein. For example, the steps recited in any of the method or process descriptions may be executed in any order and are not limited to the order presented. It will be appreciated that the following description makes appropriate reference to not only to the steps depicted in Figures 3–4, but also to the various system components as described above with reference to Figure 1 and 2. Further, illustrations of the process flows and the descriptions thereof may make reference to user windows, webpages, websites, web forms, prompts, etc. Practitioners will

appreciate that the illustrated steps described herein may comprise in any number of configurations including the use of windows, webpages, web forms, popup windows, prompts and the like. It should be further appreciated that the multiple steps as illustrated and described may be combined into single webpages and/or windows but have been expanded for the sake of simplicity. In other cases, steps illustrated and described as single process steps may be separated into multiple webpages and/or windows but have been combined for simplicity.

[Para 55] Figure 3 is a flow chart illustrating an exemplary method for classifying an asset and assigning it to an appropriate security domain. A classification of an asset may include, for example, whether the asset is public in nature, business confidential, private or secret. Such classification of an asset may determine what levels of security to apply to the asset as well as whether encryption of data may be required.

[Para 56] A second tier of information regarding an asset may comprise a business value (step 315). An asset's business value may relate to the value the asset provides in relation to its utility. For example, a data asset relating to product information which is included within a company's website, may be assigned a business value that is lower than a value assigned to data relating to accounts payable. An asset business value may be thought of in terms of, what will the resultant loss in terms of revenue and reputation if an asset is lost? For example, if a credit card issuer cannot provide its customers with an online option to pay their credit card bills due to a temporary loss of an account data asset, it may experience a minor loss in customer loyalty. However, if the credit card issuer experiences a loss in a purchase authorization asset, the issuer may experience a significant loss in revenue along with a possible loss of customers. Therefore, determining an asset business value (step 315) may be important in situations where a provisioning engine 105 must determine which assets will receive provisioning priority when resources are limited.

[Para 57] A third tier relating to an asset may comprise a resource classification for the requesting entity (step 320). A requesting entity may be a

user or another software and/or hardware component. A resource classification may be used by the provisioning engine 105 to determine what security controls to apply to an asset. For example, an internal client 110 may be considered a trusted entity, and therefore, the internal client is only required to enter a user ID and password. An external client 105 may be considered a non-trusted entity and additional security indices may be required, such as name, address, telephone number and the like. Decisioning based on a requesting entity, for example, may be determined by the originating access channel or by a digital certificate proving the source is legitimate and is from a trusted source.

[Para 58] Based on multi-dimensional decision criteria (i.e. classification of an asset, asset business value and resource classification), an asset may be placed into an appropriate security domain (step 325). Further, the multi-dimensional decision criteria may help to ensure that assets are placed with security domains providing varying levels of security controls according to domain rules which may be retrieved from a database or defined within a policy manager (step 330). Based on an assets security domain, the provisioning engine 130 may further call on various managers in order to process the request appropriately. For example, a manager may determine a geographical source or location where a transaction request originated and based on this information, determine appropriate patches and privacy rules that are applicable to the request's location.

[Para 59] Figure 4 is a flow chart illustrating an exemplary method for providing access to computing assets and provisioning and de-provisioning computing and/or data resources. A request for an asset (step 400) may be transmitted from a server to the provisioning engine 130. A request may comprise information such as, for example, a source ID and/or digital certificate from the requesting computer, transaction type, asset level, classification level, workload priority, workload volume, process time duration and the like. Prior to allowing a connecting client access to a computing resource, a provisioning engine 130 may invoke a connection manager 210 to determine the fitness of the connecting entity (step 402). In order to reduce

the risk of infecting and damaging enterprise resources with malicious code (i.e. bugs and viruses), a risk manager 210 may evaluate the connecting client in order to verify that appropriate patches and/or antivirus software is detected. If a risk manager 210 detects that the "health" of a connecting client is not in accordance with pre-determined policy or that a software update or patch is required (step 405), the connection manager 225 may direct a user to download a software update and/or patch (step 410).

[Para 60] If the appropriate software and/or patch has been downloaded and installed (step 410) within the connecting client, and/or if the connecting client was found to be compliant with risk manager 210 requirements, the provisioning engine 130 may invoke a policy manager 215 to determine which data policies to apply to an asset based on its security domain. For example, if during the classification of an asset (Figure 3, step 310) it is determined that the asset relates to private or secret information, then the provisioning engine 130 may require the asset to be encrypted. If it is determined that an asset requires data encryption (step 420), then the provisioning engine may invoke an encryption procedure (step 425).

[Para 61] The provisioning engine 130 may invoke a configuration manager 220 in order to set asset configuration settings based on its security domain (step 430). Configuration settings may maintain appropriate settings regarding which services a server may restrict. For example, if the asset is in a security domain indicative of a very critical asset, the configuration manager 220 may instruct the provisioning engine 130 to turn off all services that are not absolutely necessary. This may ensure that the asset has all of the resources necessary. When the asset lifecycle has expired, previously restricted services may be reinstated.

[Para 62] Following retrieval of settings from the various manager components, the provisioning engine 130 may interface with a provisioning tool 235 in order to provision and/or de-provision resources (step 435). As previously described, a provisioning tool 235 may comprise any commercially available resource allocation or load-balancing tool. The provisioning tool may be thought of as a switching station. The provisioning tool 235 may be

configured to receive instructions and act accordingly without applying its own logic, thereby relying on provisioning and de-provisioning instructions as received by the provisioning engine 130. When resources have been provisioned in compliance with the asset's security domain, the provisioning process may be complete (step 440).

[Para 63] The steps as recited above and illustrated in Figures 3 and 4 provide a sample process flow for an exemplary embodiment for dynamic provisioning, practitioners will appreciate the steps may be executed in any order. Additionally, other steps and additional parameters could be added to the process while others may be removed without departing from the spirit of the invention.

[Para 64] Benefits, other advantages, and solutions to problems have been described above with regard to specific embodiments. However, the benefits, advantages, solutions to problems, and any element(s) that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as critical, required, or essential features or elements of any or all the claims. As used herein, the terms “comprises”, “comprising”, or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. Further, no element described herein is required for the practice of the invention unless expressly described as “essential” or “critical”.

[Para 65] It should be understood that the detailed description and specific examples, indicating exemplary embodiments of the present invention, are given for purposes of illustration only and not as limitations. Many changes and modifications within the scope of the instant invention may be made without departing from the spirit thereof, and the invention includes all such modifications. Corresponding structures, materials, acts, and equivalents of all elements in the claims below are intended to include any structure, material, or acts for performing the functions in combination with

other claim elements as specifically claimed. The scope of the invention should be determined by the appended claims and their legal equivalents, rather than by the examples given above.